다자간 협동 공학설계를 위한 DCWA 시스템의 설계 및 구현

이 병 걸[†]

요 약

기존의 CSCW (Computer Supported Collaborative Work) 시스템은 단순히 문서나 메시지를 구성하고 전달하는 기능만을 지원할 뿐 협동 작업자들간의 조직구성, 업무분할 및 통합, 작업흐름 제어와 같은 인지학적 기능 등은 효과적으로 반영하지 못하고 있다. 본 연구는 협동작업 환경에서의 CAD (Computer Aided Design)/CASE (Computer Aided Software Engineering)와 같은 공학 디자인 프로세스를 인지학적으로 지원하기 위한 CSCW 시스템을 개발하는데 그 목적이 있다. 제안된 DCWA (Distributed Collaborative Writing Aid) 시스템은 CAD/CASE 및 시뮬레이션 기능을 하나의 시스템으로 통합함으로써 조직구성, 업무분할, 작업흐름제어가 이들 기능에 통일적으로 적용될 수 있도록 하였다. 특히 CAD도구에서는 다중 사용자들 소유의 그림객체 간의 바인딩, 어태치먼트, 스케일링 기능들은 제공하여 결합 형태 및 정도를 나타내는 관계성 표현이 가능하도록 함으로써 업무분할 및 병합이 CAD 도구에서도 명확히 드러나도록 하였고, CASE 도구에서는 소프트웨어 시뮬레이션 기능과 연계하여 작업흐름 제어가 용이하도록 하였다. 이는 또한 제품을 생산하기 전에 미리 모형을 시뮬레이션 해봄으로써 제품 개발의 시간과 비용을 단축할 수 있는 효과를 가져다 준다.

Design and Implementation of Distributed Collaborative Writing System for Engineering Design Process

Byong Gul Lee[†]

ABSTRACT

Most work done in CSCW (Computer Supported Collaborative Work) system has been targeted toward supporting the exchange of documents or messages among group members, and yet support for cognitive aspects such as group organization, division and merge of work, and work flow control. The objective of the study is to provide CSCW environment for the engineering design process such as CAD (Computer Aided Design) and CASE (Computer Aided Software Engineering). The proposed DCWA (Distributed Collaborative Writing Aid) system suggests a mechanism that unifies the group organization, work division, and work flow control in the CAD, CASE, and software simulation tool. In particular, CAD relates the group and work partition by providing expressing the relation of drawing object (e.g., binding, attachment, and proportional scaling) which is owned by different members of group, and CASE combining with the simulation tool supports the flexibility of the work flow control. Simulating the prototype before manufacturing a product can reduce time and cost in development.

1. 서 론

지금까지 대부분의 컴퓨터 소프트웨어의 개발은 단일 사용자 환경 중심으로 이루어져 왔다. 하지만 컴퓨터 기술의 발달과 더불어 문제해결의 영역 또한 커지고 복잡해짐에 따라 단일 사용자 환경만으로는 주어진 과제나 업무를 쉽게 해결할 수 없는 경우가 발생하기도 한다. 이에 다중의 사용자들이 시간적 공간적 제약을 받지 않고 컴퓨터를 이용한 협동작업을 통하여 주어진 문제를 해결하고자 하였고, 90년대에

† 서울여자대학교 컴퓨터학과 조교수

이르러 급속히 확산된 네트워크 및 인터페이스 기술은 이러한 변화를 더욱 가속화 시키고 있다. 다중 사용자들의 협동작업을 돕기 위한 시스템 개발에 관한 연구를 CSCW (Computer Supported Collaborative Work) 라고 하며 이미 전자문서결재 시스템이나 가상강의 시스템과 같은 분야에서 그룹웨어의 형태로 개발되어 사용되고 있다[1]. 하지만 그룹웨어 시스템에 이용되는 기존의 기술들은 단순히 문서나 메시지를 구성하고 전달하는 기능만을 지원할 뿐 협동 작업자들간의 상호성을 바탕으로 한 조직구성, 업무분할 및 작업흐름 제어와 같은 인지학적 기능 등은 효과적으로 반영하지 못하고 있다. 예를 들어, 협동작업 하에서의 CAD/CASE와 같은 공학설계 프로세스는 일반적으로 토론, 협동 및 시뮬레이션 과정을 반복적으로 순환하는데, 이때 요구되는 인지학적 특성은 각 과정별로 적용되는 형태나 범위에 따라 다르게 나타날 수 있다. 각 과정은 또한 문제 인식 및 분석 단계, 브레인스토밍 단계, 디자인 단계, 시뮬레이션 단계, 모형개발 단계, 생산 단계와 같은 세부 단계를 차례로 거치게 되며 각 단계별로 다양한 인지학적 요소들이 선택적 또는 반복적으로 적용된다[18]. 각 단계에서 나타나는 인지학적 요소들 중 업무조정 및 분할 기능은 주로 문제 인식 및 분석 단계에서 이용되고 화상회의와 같은 토론 기능은 브레인스토밍, 모형개발 단계에서 집중적으로 요구된다. 업무 흐름제어 및 다중 사용자 인터페이스 기능은 모든 단계에서 연속적으로 지원되어야 하며 디자인 단계, 시뮬레이션 단계, 모형개발 단계에서는 CAD/CASE 기능 외에도 토론 및 업무 통합기능 등이 선택적으로 지원될 수 있어야 한다. 하지만 기존의 그룹웨어 시스템은 이러한 응용 분야별 특성을 고려하지 않고 설계되어 다중 사용자간의 다양한 인지학적 행동을 만족스럽게 지원할 수가 없었다. 즉 시스템 설계에 있어 단순히 중앙 집중식 또는 분산 방식을 일률적으로 채택함으로써 업무 분할이나 통합이 용이하지 않고 협동작업 흐름의 유연성을 저하시키는 요인이 되어 왔다. 또한 기존의 협동저작 시스템에서는 시뮬레이션 기능이 지원되지 않아 설계된 모형을 직접 테스트하는데 많은 비용을 지출해야 하는 문제점이 있었다. 예를 들어 원거리 통신 네트워크 시스템이나 비행기의 비상탈출 시스템 등은 모형의 구현과 테스트에 전체 개발비용의 대부분을 소요하고 있는 실정이다. 따라서 모형개발을 위한 모델링 기능과 모형의 테스팅을 위한 시뮬레이션 기능이 서로 연계 통합 된다면 시스템 개발 및 통합에 요구되는 비용을 크게 절감할 수 있을 것이다.

본 논문에서는 협동작업 환경에서 CAD/CASE와 같은 공학디자인 프로세스를 효과적으로 지원하기 위하여 요구되는 업무 분할 및 통합, 업무 흐름제어, 토론, 시뮬레이션 기능과 같은 요소들을CSCW 시스템에 통합하기 위한 방법 및 모델을 제시하고 제안된 모델이 어떻게 협동CAD/CASE 디자인 작업을 효과적으로 지원할 수 있는지를 기술한다. 2절에서는 협동 CAD/CASE 디자인을 지원하기 위한 분산 협동작업 시스템(DCWA)의 목적 및 관련 연구에 대해 기술하고 3절에서는 DCWA의 주기능을 소개한다. 4절에서는 DCWA 시스템의 구조 및 구현 방법에 대해 설명하고 5절에서는 DCWA를 CAD/CASE 작업에 적용한 구현사례를 소개한다. 마지막으로 6절에서는 결론 및 향후 연구 방향을 기술한다.

2. 관련연구

기존의 협동환경에서의 컴퓨터는 단지 데이터를 수집, 분석, 저장하는 수동적인 역할만을 담당해왔으며 그룹 구성원간의 상호이해, 업무조정 및 의사결정 등은 그룹 구성원들의 책임이었다. 이러한 환경은 점차 컴퓨터가 그룹 구성원간의 협동행위 조정에 능동적으로 개입하는 환경으로 바뀌어져 가고 있으며, 현재 이를 반영한 협동저작 시스템의 개발이 활발히 진행되고 있다[22].

MERMAID[19]와 CoSARA[20]는 텍스트, 음성, 영상을 통한 실시간 그룹토론을 지원하기 위한 그룹 컨퍼런싱 시스템으로서 그룹 구성원들의 다양한 아이디어를 수집하고 정형화함으로써 브레인스토밍 기능까지도 지원한다. 특히 CoSARA는 실시간 그룹 컨퍼런싱 시스템 구축을 위한 개발환경을 지원하기 위하여 Object World (OW) 개념을 사용한다. 즉 시스템 개발자는 다양한OW 객체를 이용하여 원하는 어플리케이션을 작성하거나 재 구성할 수 있다. 따라서 OW는 객체 정보의 공유 상태 (degree) 및 크기 (granularity)를 결정하기가 용이하다는 장점이 있다. 하지만 OW는 LISP 언어로 구현되어 있기 때문에 이를 호출할 어플리케이션도 LISP로 재 작성해야

하는 불편함과 인터프리터 방식 언어의 단점인 성능 저하를 초래할 수 있다.

널리 알려진 rIBIS와 SEPIA[7]와 같은 텍스트 기반 협동저작 도구는 다중사용자를 위한 인터페이스 기능을 갖추고 있으며 다양한 협동모드를 지원한다. 독립모드 (independent mode)에서는 사용자가 상대방의 간섭 없이 독자적으로 업무를 수행할 수 있으며, 느슨한 관계 모드 (loosely-coupled mode)에서는 사용자들간의 전역 정보만을 서로 공유할 수 있게 하며 밀접한 관계 모드 (tightly-coupled mode)에서는 병행처리 시 생길 수 있는 충돌을 방지하기 위하여 다중사용자간에 파일이나 데이터베이스에 대한 동시 쓰기를 허락하지 않는다. 이 경우, 읽기는 허락함으로써 사용자들 간에 동시에 같은 관찰 뷰를 공유할 수 있도록 한다. 하지만 이들 시스템에서는 업무조정이나 흐름제어와 같은 인지학적 기능은 지원하지 못한다.

TeamWorkStation (TWS)[4]에서는 모든 사용자가 동시에 다양한 그림도구를 사용하여 실시간으로 그림 작업을 할 수 있는 그림 저작 도구이다. 또한 여러 사용자에 의해 만들어진 그림 이미지의 중첩이 가능하도록 오버레이 (Overlay) 기능을 지원한다. 하지만 TWS와 같은 CAD 도구는 그림 객체들간의 관계성과 소유자를 명시하지 않기 때문에 하나의 그림 객체를 변경할 경우 다른 그림객체 들과의 위치좌표 및 소유자 정보를 함께 잃어 버리는 단점이 있다. 이러한 정보의 손실은 잘못된 정보를 공유함으로써 협동저작 행위에 장애를 줄 수 있으며 작업의 생산성을 저하 시키는 요인이 된다.

GroupPaint[5]는 클라이언트-서버 모델에 기반한 그림 저작도구로서 그룹웨어 서버, 어플리케이션 서버 및 클라이언트로 이루어 지며 마이크로 소프트사의 Paintbrush에서부터 발전되어 왔다. 따라서 모든 그림 정보들은 비트맵 형식으로 저장되고 조작되어진다. 협동작업은 이들 비트맵을 중첩함으로써 이루어 진다. 하지만 GroupPaint 역시 그림 객체들간의 관계성 명시가 불명확할 뿐 아니라 구성원들간의 의사소통을 위한 효과적인 통신수단을 지원하지 못하고 있다.

X 윈도우 기반의 Ensemble[6]에서는 그림 객체들의 병행처리를 위해 묵시적 잠금 (Implicit Locking)을 사용함으로써 다른 사용자가 이미 잠금된 객체에 대한 접근을 하지 못하도록 막아준다. 하지만 Ensemble에서는 상대방의 행동이나 모습을 관찰할 수 있는 화상회의 기능이 없으며 편집 진행중인 그림객체도 실시간으로 관찰할 수 없다는 단점이 있다. 이는 그룹 구성원들간의 효과적인 의사소통을 방해함으로써 생산성을 저하시키는 요인이 된다. 또한 GroupPaint에서와 같이 그림 객체들간의 관계성 명시가 불명확하다.

ClearCase[21]는 텍스트 기반의 CASE 도구로서 다중 사용자로 하여금 다양한 어플리케이션을 개발할 수 있는 기능을 지원한다. 특히 ClearCase에서 지원되는 프로젝트 관리 기능은 관리자가 구성원들간의 업무를 분담하고 할당하거나 각 구성원들의 업무 진행사항의 파악을 용이하게 해준다. 하지만 구성원들간의 협동행위가 실시간으로 이루어지지 못하며 다른 사용자의 파일을 변경하거나 삭제할 수 있는 접근 허가 기능이 없다. 이는 실시간 시뮬레이션의 구현에 장애 요소가 될 수 있다.

요약하면, 기존의 협동 CAD/CASE 도구는 정보의 관찰 및 공유를 제어하는 온라인 조정기능을 지원하지 못하거나 극히 제한된 기능만을 지원하고 있다는 공통점이 있다. 또한 대부분의 시스템들이 모형 테스트를 위한 소프트웨어 시뮬레이션 기능을 제공하지 않음으로 인해 CAD/CASE 도구를 협동 공학 디자인 프로세스에 효과적으로 이용할 수 없는 실정이다[23].

DCWA는 기존의 일반적인 협동작업 시스템을 협동 CAD/CASE 디자인 작업에 적용하기 위하여 개발된 응용 시스템이다. 따라서 협동 CAD 도구는 지리적으로 분산된 사용자들이 공동으로 그림이나 도면을 그릴 수 있는 기능을 제공해야 하며 분산된 그림객체의 통합을 위하여 그림 병합기능이 지원되어야 한다. CASE 도구는 사용자들이 시뮬레이션 코드를 시간적, 공간적 제약 없이 자유로이 작성 할 수 있도록 지원해야 하고, 독립적으로 작성된 코드 조각을 통합하여 하나의 실행프로그램을 만들 수 있는 기능을 제공하여야 한다. 또한 분산된 사용자들이 네트워크를 통하여 상호간 토론을 진행하거나, 텍스트나 그림 및 프로그램등을 저작하면서 동시에 상대방의 저작 행위를 관찰할 수 있도록 기존의 단일사용자 환경의 그래피컬 인터페이스를 다중사용자 관점에서 재 디자인 하여야 한다. 이때 화상 및 영상 커뮤니케이션과 멀티미디어 데이터베이스가 함께 지원되

어야 한다.

3. 시스템의 주 기능

3.1 분산 CAD 도구

분산 협동 CAD 도구에서는 다양한 형태의 그림 객체들이 결합될 수 있어야 한다. 디자이너는 결합이 어떻게 완성되어야 할 지에 관한 토론을 이끌어야 하며 이러한 토론은 실제로 디자인을 시작하기 전에 선행되어야만 한다. 그렇지 않으면, 지향하는 디자인의 목적을 잃어버리기 쉽다. 이를 위하여 서로 다른 그림객체의 결합을 책임지는 온라인 결합 기능이 필요하다. 협동 저작을 지원하기 위한 몇몇CAD 모형들이 개발되어 왔음에도 불구하고 그들 대부분은 온라인 결합 기능이 빠져 있다. 예를 들어, 대부분의 시스템들은 그림객체의 중첩기능은 지원하지만 바인딩, 어태치먼트, 또는 스케일링과 같은 결합 기능들은 제공하지 않는다. 제안된 DCWA시스템에서는 그림 객체들간의 결합 형태 및 정도를 나타내는 관계성 표현이 가능하다. 이런 관계 정보는 데이터 베이스에 저장되고 진행중인 모든 협동 참여자의 그림 객체들과 결합하기 위해 온라인 결합자에게 제공된다.

3.2 CAD와 CASE의 결합

제안된 컴퓨터 협동 저작 시스템은 그룹 시뮬레이션 환경에서도 이용될 수 있다. 각각의 그룹 구성원은 다른 구성원이 작성한 프로그램 코드를 볼 수 있으며 다른 구성원이 작성한 프로그램 코드와의 결합도 가능하다. 프로그램 조각들을 결합하여 하나의 완성된 프로그램으로 만드는 작업은 그룹의 각 구성원들의 책임으로 남겨진다. 또한 공학 디자인 프로세스에서는 실제로 제품 디자인과 시뮬레이션 디자인이 병렬적으로 이루어진다. 따라서 이 두 프로세스의 반복적 수행을 피하기 위하여 CAD 도구와 CASE 도구의 결합이 필요하다. CAD 및 CASE 도구의 결합은 공학디자인 각 과정에서 나타나는 인지학적 요소들을 고려하여 사용자가 동일한 환경에서 각 도구들을 사용할 수 있도록 설계되어야 한다.

3.3 커뮤니케이션과 디자인 행위의 동기화

기존의 CSCW 시스템들은 데이터, 음성, 비디오 정보들을 각각 독립적 전송채널로 운용해 왔다. 예를 들어, 대부분의 멀티미디어 화상회의 시스템에서 음성정보는 화상정보와는 독립적으로 전송되는데, 협동작업 환경에서는 이러한 복수 매체의 정보의 흐름을 조절하기 위한 동기화 메커니즘이 필요하다[2]. DCWA에서는 타임스탬프를 이용하여 다양한 멀티미디어 정보를 동기화 한다. 특히 타임스탬프를 이용하여 사용자의 음성 및 화상 정보들을 CAD/CASE 그림 정보들과 상호 동기화 함으로써 CAD/CASE작업의 UNDO/REDO 기능 구현 시 그림객체만 복귀되는 것이 아니라 관련 시간프레임 내에서의 음성 및 화상정보도 복구할 수 있도록 하였다.

3.4 그룹 관리

CSCW 시스템 사용자가 가장 처음으로 접하는 기능으로서 그룹의 생성 및 보안 기능을 지원한다. 이를 위하여 DCWA는 그룹서버를 두고 있으며 그룹서버에는 그룹이름, 구성원 그룹리더, 활동디렉토리 등의 정보를 저장한다. 특히 한 구성원이 여러 개의 그룹에 참여 할 때 생기는 보안 문제를 해결하기 위해서 유닉스의 그룹 접근 기능을 도입하였다. 또한 각 그룹에 할당되는 활동디렉토리 개념을 유닉스 그룹 기능과 연계하여 각 파일의 접근 권한을 그룹에 따라 달리 부여할 수 있도록 하였다. 그룹책임자는 그룹이름 및 구성원의 조직을 책임지며 그룹구성원은 그룹책임자의 승인하에 그룹참여를 허가 받는다. 그룹책임자는 또한 그룹 구성원간의 합의를 통하여 업무를 분담하며, 지향하는 문서의 구조 및 문서 형태 (e.g., 텍스트, 그림, 코드)를 결정한다. 각 문서의 생성, 편집, 분할, 결합, 시뮬레이션은 그룹구성원 각자의 책임하에 이루어진다.

4. DCWA의 구조

4.1 DCWA

DCWA는 SOLARIS 2.5, TCP/UDP/IP 프로토콜, OSF의 MOTIF 툴킷, X라이브러리 및 C++과 같은 프로그래밍 표준을 채택하여 시스템 이식성이 뛰어나다. DCWA는 크게 협동관리 시스템, 분산 멀티미디어 데이터 베이스, 다중 사용자 인터페이스 및 소프트웨어 시뮬레이션 기능으로 나누어 진다. 협동관

리 시스템은 그룹의 협동 환경을 지원하기 위하여 협동제어패널, 협동 데이터베이스, 세션 관리 및 협동 조정자와 같은 네 기능으로 분류된다. 분산 멀티미디어 데이터 베이스는 텍스트 및 그래픽 정보를 구조적 관계성을 바탕으로 관리하며 사용자는 이러한 정보의 구조적 구성을 세만틱 망 (Semantic Network) 을 사용하여 유기적으로 구성할 수도 있다. 사용자 인터페이스는 사용자로 하여금 자신의 문서 뿐 아니라 다른 사람의 문서 까지도 WYSIWIS(What you See Is what I see) 형식으로 관찰 할 수 있게 해주며 CAD/CASE 편집 및 검색 도구 등을 지원한다. 또한 DCWA는 다중의 사용자가 공동으로 소프트웨어 개발에 참여하여 모형 시스템을 미리 만들어 테스트 해볼 수 있도록 소프트웨어 시뮬레이션 기능을 지원한다. 이를 위하여 CASE 도구에서 생성된 코드 조각들을 실행 가능한 파일로 통합하여 컴파일링 할 수 있는 기능을 지원한다. 그림 1은 DCWA각 모듈의 제어 및 데이터 흐름을 보여준다.
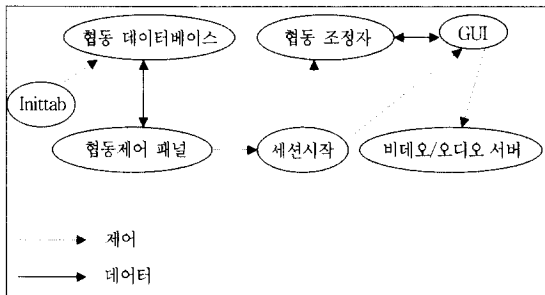
그림 1. DCWA의 모듈 구조

4.2 협동관리 시스템

4.2.1 협동 데이터베이스 (Collaboration Database Management System)

협동 데이터베이스는 호스트 상에서 실행되는 데몬 프로세서로써 모든 지역 망에 연결되어 있는 사용자들로부터의 접근이 가능하다. 데몬 프로세서는 사용자가 DCWA를 시작 할 때 새로운 협동 작업 세션을 생성하거나 기존의 협동작업 데이터베이스를 갱신 또는 삭제할 수 있게 한다. 이 데몬 프로세서는 INITTAB프로그램에 의해서 유지 되는데, 만약 데몬 프로세서가 정지되면 INITTAB 이 데몬 프로세서를 자동적으로 재 시작 시킨다.

4.2.2 협동 제어 패널 (Collaboration Control Panel)

사용자가 새로운 협동 세션을 만들거나 기존의 협동 세션을 시작 하고자 할 때, 협동 제어 패널은 요구 메시지를 협동 데이터베이스에게 보내고 협동 데이터베이스로부터의 결과 메시지를 수신하는 역할을 담당한다. 협동 데이터베이스는 사용자의 요구에 따라 자신의 데이터베이스를 갱신하고 결과를 협동 제어패널에 돌려준다. 다음으로 협동 제어패널은 협동 세션에 접근하기 위한 허가권을 검사하고 협동 디렉토리를 생산한다. 그림 2와 그림 3은 새로운 협동 세션을 생성하고 시작하는 윈도우를 보여준다.

4.2.3 세션 시작 (Session Start)

세션 시작 도구는 세션 조정자 실행기와 그래피컬

그림 2. 새로운 그룹의 시작 창

그림 3. 기존 그룹의 선택 창

사용자 인터페이스 실행기로 나누어 지는데 세션시작은 새로운 협동이 생성되었을 때 또는 사용자가 기존의 협동 세션을 시작하고자 할 때 불려 진다. 세션 시작 도구는 협동 조정자가 이미 실행되어 있지 않음을 발견하면 새로운 조정자를 생성한다.

4.2.4 협동 조정자 (Collaboration Coordinator)

협동 조정자는 시스템의 병렬 제어를 지원하기 위한 서버 프로세서이다. 협동 조정자가 처음 시작될 때 시스템의 '.active' 파일을 사용하여 사용자 이름과 네트워크 소켓 정보를 파일에 저장 한다. 네트워크 연결요구가 도착 했을 때 소켓과 포트를 열고 호스트와 클라이언트간의 커뮤니케이션을 지원한다.

4.3 다중사용자 인터페이스

다중사용자 인터페이스는 협동 작업환경을 위한 문서의 논리적 구성 (Logical View), 문서 편집, 저장, 통합 기능등을 제공한다.

4.3.1 논리 뷰 편집기

DCWA에서는 각각의 문서가 하나의 논리적 단위로써 비중첩적으로 분할된다. 각각의 논리적 단위들은 노드로 표현되며 이러한 노드들은 비순환성 비가중 나무구조로 배열된다. 논리 뷰는 사용자로 하여금 문서의 다른 조각들을 동시에 선택하고 편집할 수 있도록 해 준다. 이러한 기능은 공유정보의 개인적 관찰 기능을 가능하게 한다. 협동 문서 저작 환경에서 개인적 관찰 기능은 커뮤니케이션 부하를 현저히 줄일 수 있다. 또한 논리적 구성 뷰에서는 노드 삽입 및 삭제기능을 지원한다. 그림 4는 논리 뷰를 포함한 메인 윈도우를 보여준다. 이 구조의 설계는 각 그룹 구성원의 책임으로 이루어진다. 다중의 사용자가 동일한 노드를 편집할 때 생길 수 있는 충돌은 간단한 잠금 기능을 제공 함으로써 예방 할 수 있다. 현실적으로 모든 사용자가 협동 작업을 마칠 때 까지 동시에 어느 특정 노드를 쓰거나 편집할 확률은 적다. 이 사실은 간단한 잠금 메커니즘이 협동 작업의 성능을 그다지 심각하게 저하시키지는 않는다는 것을 증명해 준다[17]. 이외에도 각각의 노드는 노드 이름, 노드 타입, 노드 소유자, 파일 이름, 제목, 키워드와 같은 속성을 지원한다.

Collaboration: dissertation; Group: csemm
File View Search Document Case Videoconference Help
dissertation
Introduction
Intro
Taxonomy
Features
Issues
Sy
Successfully connected to server

그림 4. 논리뷰

4.3.2 텍스트 및 그래픽 편집기

텍스트 편집기는 프로그램을 시뮬레이션하거나 텍스트를 저장 편집하는데 사용되며 그래픽 편집기는 그림을 그리거나 문서에 그림요소를 삽입할 때 사용될 수 있다. 사용자가 논리 창에서 특정 노드를 선택 할 때 선택된 노드의 내용에 따라 텍스트 또는 그래픽 편집기가 불려진다. 그림 5는 DCWA의 텍스트와 그래픽 편집기를 보여준다.

DCWA Text Editor: About.nod
DCWA Drawing Editor: Drawing.dgf

그림 5. 텍스트 및 그래픽 편집기

그래픽 편집기의 CAD 기능은 이차원 그림도구 기능이 지원되며 가변 및 고정 스케일링 기능과 같은 다양한 객체 결합 기능을 갖추고 있다. 기존의 픽셀

기반의 그래픽 편집은 중첩 병합은 지원하지만 객체들 간의 결합 관계를 정의 하지는 않는다[4-6]. 객체들 간의 결합 필요성은 다음의 주택 설계 예에서 보여진다. 하나의 주택설계는 바닥, 배수, 전기, 냉난방 설계를 포함한다. 디자인 팀은 각각의 설계업무를 할당 받아 디자인 작업을 수행하게 되는데, 완성된 디자인은 먼저 주 그림을 배경에 위치하고 다양한 서브 객체들을 그 위에 결합 함으로써 얻어질 수 있다. 이때 주 그림이 변경될 때 마다 주 그림상의 다른 객체들 또한 변경되어야 한다[18]. 그래픽 편집기는 또한 그래픽 객체를 포스트 스크립터와 같은 다른 형태로 전환 표시 할 수 있으며 텍스트 및 그래픽 객체를 하나의 문서로 결합할 수 있게 해 준다.

4.3.3 문서 통합

DCWA는 대용량의 문서를 작성하기 위한 공동저작 환경을 제공한다. 각각의 구성원은 그 자신의 업무 뿐 아니라 동시에 다른 사람의 업무를 관찰할 수 있어야 한다. 만약 사용자가 전체 문서를 보거나 출력하고자 할 때 사용자는 논리적 문서 구조로부터 하나의 완성된 문서 구조를 생성할 수 있어야 한다. 이를 위하여 DCWA에서는 두 가지 옵션이 제공된다. 주 윈도우의 메뉴에서 사용자는 문서 생성기 메뉴를 선택함으로써 전체 문서나 재단된 문서를 생성할 수 있다. 만약 사용자가 전체 문서를 생성하기를 원한다면 사용자는 단순히 'generate whole document' 옵션을 선택할 수 있다. 재단된 문서를 생성하기 위해서는 사용자가 포함되어야 할 문서조각 노드들을 선택하고 재배치 할 수 있어야 한다. 사용자는 먼저 노드의 내용을 확인하고 문서 구조를 결정할 노드들을 선택한다. DCWA는 이것을 취합하여 재단된 문서를 생성한다.

4.4 데이터베이스

4.4.1 검색도구

문서의 크기가 증가됨에 따라 문서의 내용이나 노드수의 증가로 인한 문서의 논리적 구성이 복잡해 질 수 있다. 따라서 사용자로 하여금 원하는 문서를 쉽게 검색할 수 있게 하게 하기 위한 검색도구가 필요하다. 이러한 검색도구는 다른 사람에 의하여 작성되어진 문서의 내용을 미리 인지할 필요가 있을 때 이용될 수 있다. 예를 들어, 특정 노드가 C프로그램 코드를 포함하고 있을 때 사용자는 코드의 내용을 이해하기 위하여 코드 작성자에게 그 내용을 문의 할 수 있다. 이때 제안된 검색도구는 노드들의 주어진 속성을 이용하여 다른 사용자들의 작업 정보를 쉽게 얻을 수 있게 한다[11]. 검색 도구를 사용하기 위하여 사용자는 질의어를 사용할 수 있으며, 질의 조건을 만족하는 결과 노드들은 윈도우에 표시된다.

제안된 검색 도구는 노드 속성을 선택할 수 있는 기능을 제공한다. 예를 들어 사용자는 이름과 키워드 같은 특정한 속성을 이용하여 노드들을 검색 또는 선택할 수 있다(그림 6). 가끔 노드 속성만으로는 사용자에게 충분한 정보를 제공하지 못하는 경우가 있다. 예를 들어 C프로그램을 위한 사용자 도움말을 작성할 때 도움말 노드는 다양한 형태의 오류 메시지를 포함하는 몇 개의 자식 노드로 분기 될 수 있다. 논리적 구조에 근거한 검색 도구는 이러한 노드들 간의 의미적 관계를 정의 할 수 없다. 세만틱 네트워크의 기능을 사용하면 이러한 노드들 간의 구조적 관계를 의미적 관계로 재 정의할 수 있는 장점이 있다.

그림 6. 탐색 창

4.4.2 세만틱 네트워크

대부분의 CSCW도구들은 각각의 노드에 레이블을 부여할 수 있는 기능을 제공한다. 그러나 하나의 레이블이 포함할 수 있는 정보의 양은 제한되어 있다. 이를 개선하기 위하여 노드의 속성과 해당 값을

기술하는 기법이 제안된다. 이들 속성과 값이 주어지면 논리적 구조의 탐색이 가능하다. 이들 속성과 값은 논리적 구성 이외에도 세만틱 네트워크를 이용하는데 사용될 수 있다. 즉, 각 노드들을 일정한 형태의 상속구조로 관계 지울 수 있다[11].

예를 들어 C프로그래밍에서 특정 노드가 도움말 메시지에 관한 내용을 담고 있으며, 도움말 노드는 다시 몇 가지의 오류 메시지 노드와 오류 해결 노드로 분기된다고 하자. 이때 도움말 노드의 속성과 값은 'Contains/Errors', 'Contains/Trobleshooting'으로 각각 표현될 수 있다. 한편, 도움말이 전체 문서의 한 부분이라고 한다면 'IS_a/Document' 값이 생성될 수 있다. 사용자가 탐색 도구를 이용하여 먼저 도움말 노드를 찾은 후에 세만틱 네트워크 기능을 사용함으로써 도움말에 대한 세만틱 네트워크를 구성할 수 있다. 결과가 그림 7에 보여진다. 그림에서 색칠된 버튼이 사용자가 질의한 결과의 노드들이다. 부모와 자식 노드들이 질의 노드 주변에 배치됨을 볼 수 있다. 이런 관계성은 더 이상의 노드가 발견되지 않을 때 까지 계속 확장된다. 만약 사용자가 세만틱 네트워크상의 특정 노드의 내용을 보기를 원한다면 사용자는 단순히 일반 노드 브라우징에서와 같이 원하는 노드를 클릭하기만 하면 된다. 이처럼 각각의 노드를 속성/값 쌍으로 관계 지움으로써 세만틱 네트워크를 노드의 논리적 구성과 함께 표현할 수 있고, 사용자 관점의 새로운 뷰를 제공할 수 있다[13,14].

그림 7. 세만틱 네트워크

4.5 실시간 커뮤니케이션

실시간 커뮤니케이션 시스템 구현을 위해서는 음성 및 화상 객체들간의 동기화 기능이 매우 중요하다[8,9]. 이런 조절기능의 목적을 달성하기 위하여 DCWA는 타이밍 인터페이스 및 프로토콜을 지원한다. 타이밍 인터페이스는 음성 및 화상 서버간의 통신 소통의 시간 관계성을 정의하기 위한 메커니즘으로, 고려 되어야 할 사항으로써 1) 얼마나 자주 화상 및 음성 데이터가 동기화 되어야 하는가? 2) 비동기 현상이 발생했을 때 어떤 데이터가 취사선택 되어야 하는가? 등이 결정되어야 한다[3,10]. 타이밍 프로토콜의 동기화를 구현하기 위해서 1) 각각의 패킷에 타임스탬프를 부여하거나 2)도착한 패킷을 적당히 버퍼링 할 수 있는데 DCWA에서는UDP/IP 프로토콜에서 타임스탬프를 부여하는 방법을 채택하고 있다. 화상 회의를 시작하기 위하여 먼저 음성 및 화상 서버와 클라이언트 관리자가 실행된다. 서버는 원격지의 클라이언트에게 데이터를 전송하고 클라이언트 관리자는 원격지 서버로부터 데이터를 받는다. 연결 서버는 각 클라이언트로부터의 연결 및 비연결 요구를 처리한다. 연결요구가 도착했을 때 연결 서버는 새로운 포트 넘버를 보내고 사용자가 통신을 시작할 수 있도록 새로운 포트에 연결한다. 그림 8은 화상 회의 시 나타나는 화상 이미지를 보여준다.

그림 8. 화상회의 화면

4.6 소프트웨어 시뮬레이션

시뮬레이션 기능을 제공하기 위하여 새로운 형식의 노드, 즉 시뮬레이션 노드가 필요하다. 시뮬레이션 노드들이 생성된 후 DCWA는 이들을 통합하여 실행 코드를 생성한다(그림9). 이를 위하여 DCWA는 다음과 같은 기능을 지원한다. 첫째, 코드 조각들을 적당한 순서로 통합할 수 있다. DCWA에서는 노드 구조가 사용자의 자유의지에 맡겨지기 때문에 시뮬레이션 노드들이 실행 코드로 병합될 때 일관된

순서로 배치될지는 누구도 보증 할 수 없다. 단지 코드 조각들을 결합시키는 것 만으로는 실행 프로그램 흐름을 제어하기 어렵기 때문이다. 둘째로는 변수 명명규칙의 혼란을 예방하는 기능을 지원한다. 여러 사용자들이 공동의 프로젝트를 수행할 때 변수명의 충돌이 일어날 수 있다. 즉 하나의 변수가 여러 노드에서 발견될 수 있기 때문이다. 세번째로 광역 데이터 구조 정의에 필요한 기능을 지원 한다. 일반적으로 프로그래밍 작업에서는 동일한 데이터 구조가 여러 모듈에 동시에 사용될 수 있다. 이때 각각의 데이터 구조를 모든 노드에 정의하는 것은 바람직 하지 못하다. 따라서 이러한 공유 데이터 구조는 한 노드에 한번만 정의하고 다른 코드 노드들은 이것을 사용하기만 하면 된다. 마지막으로 DCWA에서는 프로그램을 잘게 나누어 나무구조를 형성함으로써 변경된 파일만 컴파일 할 수 있도록 하였다. 이 기술을 사용하면 프로그램을 여러 파일로 나눔으로써 하나의 프로그램에 대한 동시 작업을 가능하게 하며, 작은 단위를 편집하거나 디버깅하기가 쉽고, 컴파일링 시간을 절약할 수 있다. 만약 전체 프로그램이 하나의 파일에 저장되어 있다면 프로그램 갱신이 일어날 때마다 전체 프로그램을 재컴파일 해야만 한다[15]. 결합된 코드는 협동작업의 마스터 파일에 저장되는데, DCWA는 자동적으로 이 마스터 파일을 생성하고 필요한 코드를 대체시킨다.
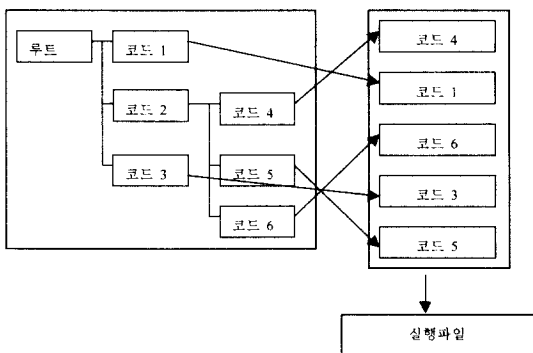
그림 9. 코드 노드와 논리 뷰와의 관계

실행 파일을 생성하기 위하여 DCWA는 MAKE 파일을 생성한다. MAKE 파일은 프로그램을 컴파일하기 위한 모든 정보를 포함하며 사용자는 이 MAKE 파일을 임의로 편집할 수 있다. 사용자는 또한 필요한 라이브러리를 사용하기 위하여 적당한 디렉토리를 임의로 생성, 사용할 수 있다. 사용자의 요구에 의하여 DCWA는 MAKE 파일을 실행하게 되는데 이때 유닉스 MAKE 유틸리티를 사용한다.

4.7 시스템 구현

DCWA의 시스템은 시스템 메시지, 오디오, 비데오, 텍스트, 그래픽스와 같은 다양한 메시지를 지원한다. 협동제어 패널과 협동 데이터베이스 또는 논리뷰 편집기 와 협동 조정자간의 커뮤니케이션은 신뢰성이 무엇보다 중요하다. 또한 이들 서브 시스템에서는 일대일 커뮤니케이션을 지원하기 때문에 브로드캐스팅이나 멀티캐스팅이 요구되지 않는다. 따라서 이들 서브 시스템에서는 TCP프로토콜을 이용하고, 비데오/오디오 컨퍼런싱과 같이 신뢰성에 민감하지 않은 시스템에서는 UDP 프로토콜을 사용할 수 있도록 설계되었다. 본 절에서는 DCWA 구조의 근간이 되는 커뮤니케이션 모델을 제시하고, 그 구현 방법을 기술한다.

4.7.1 패킷 클래스

패킷 클래스는 DCWA의 커뮤니케이션 시스템에 가장 중요한 요소가 된다. 이들 클래스는 전송 데이터를 캡슐화하여 송수신하며 수신된 데이터를 재조립하는 기능을 지원한다. 가령, 협동제어 패널과 협동 데이터베이스 간에 협동작업 정보를 전송할 경우, CDMS는 다음과 같이 협동작업 데이터를 구성한다.

```
struct Collaboration {
   string name; // the name of the Collaboration
   string group; // the name of the associated UNIX group
   string workdir; // the pathname of the working directory
   string leader; // username@domain of leader
   MemberList members; // a string list of members
   ...methods for manipulating this struct...
};
```

협동작업 데이터는 전송되기 전 하나의 데이터 스트림으로 변경되고 패킷화 된다. 다음의 CollabDataPacket 클래스가 이의 역할을 담당하며

패킷화 된 데이터는 송수신 메소드에 의해 전송된다.

```
class CollabDataPacket {
    size_t name_len; // length of name string
    size_t group_len; // length of UNIX group
    string
    size_t workdir_len;           // length of
    working directory string
    size_t leader_len; // length of leader string
    size_t num_members; // number of members
    size_t data_length; // length of string pointed
    to by data
    size_t *members_lengths; // array of member
    string lengths
    char *data; // composite string of all data
  public:
    CollabDataPacket(Collaboration *collab);
    Collaboration* extract_collaboration;
    void receive(int socket);
    void send(int socket);
    ...other supporting methods...
  };
```

```
  void CollabDataPacket::send(int socket)
  {
    iovec iov[3] = { (caddr_t)&name_len, 6*sizeof
    (size_t), (caddr_t)members_lengths, sizeof
    (size_t)*num_members, data, data_length };
    writev(socket, iov, 3);
  }
```

```
Void CollabDataPacket::receive(int socket)
  {
    read(socket, (void*)&name_len, 6*sizeof(size_t));
    members_lengths = new size_t[num_members];
    read(socket, members_lengths, sizeof(size_t)*
    num_members);
    data = new char[data_length+1];
    read(socket, data, data_length);
    data[data_length] = NULL;
  }
```

4.7.2 메시지 클래스

DCWA에서는 다양한 형태의 메시지가 교환된다. 가령, 협동 데이터베이스는 8가지 형태의 메시지를 송수신할 수 있는데, 각각은 서로 다른 형식을 가지고 있다. 하지만 협동 데이터베이스에서는 도착하는 패킷의 유형을 미리 알지 못한다. 따라서 송수신 되는 메시지의 형식을 표시하는 방법이 필요한데, DCWA에서는 이를 메시지 클래스에서 정의한다. 메시지 클래스는 3개의 독립된 메시지클래스 (협동 제어패널과 협동 데이터베이스 간의 전송, 논리 뷰 편집기와 협동 조정자 간의 전송, 그리고 비데오/오디오 컨퍼런싱 간의 전송)로 구성되며 각각의 클래스는 서비스 목적에 따라 최적화 된다 예를 들어 다음의 CDMSMessage 클래스는 협동 제어패널과 협동 데이터베이스 간의 신뢰성 있는 정보교환을 위해 이용된다.

```
struct CDMSMessage {
   CDMSMessageType type;
   timeval time; // timestamp
   void *msgdata;
   CDMSMessageType receive(int socket);
   void send(int socket, CDMSMessageType a, void
   *packet);
};
```

위의 정의에서 멤버 변수 type는 송수신되는 메시지의 형식을 나타내며 msgdata 는 송수신되는 패킷을 포인트한다. 이 메시지를 송신하기 위한 메쏘드는 아래와 같다.

```
void CDMSMessage::send(int socket,
CDMSMessageType a, void *packet)
{
   type = a;
   gettimeofday(&time, NULL);
   write(socket, &type, sizeof(CDMSMessageType)+
   sizeof(timeval));
   switch(a) {
     case ADD_COLLABORATION:
        ((CollabDataPacket*)packet)->send(socket);
        break;
     ... handle other message types ...
   }
}
```

즉, 특정 어플리케이션이 메시지를 송신할 때 먼저 메시지를 패킷화 한 후 메시지 클래스의 send 함수를 사용하여 송신한다. 메시지 클래스의 수신 메쏘드는 아래와 같다.

```
CDMSMessageType CDMSMessage::receive(int
socket)
{
   static CDMSPacketData data;
   read(socket, &type,, sizeof(CDMSMessageType)+
   sizeof(timeval));
   switch(type) {
      case ADD_COLLABORATION:
         data.collab_data_packet.receive(socket);
         msgdata = (void*)&data.collab_data_packet;
         break;
      ... handle other message types ...
   }
   return type;
}
```

receive 메쏘드는 메시지 헤더 정보를 읽고난 후 패킷 클래스의 receive 함수를 호출하여 패킷을 재조립한다. 가령 협동 데이터베이스가 ADD_COLLABORATION 메시지를 수신하였다고 한다면 다음과 같은 절차로 수신된 메시지가 패킷화된다. 따라서 DCWA 에서는 어떠한 종류의 메시지가 도착할지 미리 알 필요가 없다.

```
// determine that a message has arrived
switch(msg.receive(socket)) {
   case ADD_COLLABORATION:
      CollabDataPacket *packet = (CollabDataPacket*)
      msg->msgdata;
      Collaboration *collab = packet->extract_
      collaboration();
      // perform tests and add collab to database
      upon success
      break;
   ... handle other message types ...
}
```

5. 구현사례 및 평가

DCWA 시스템을 사용하여 다중의 사용자들이 전자우편 인터페이스를 설계하고 시뮬레이션 해보고자 한다. 먼저 그룹 책임자는 협동 관리자를 통하여 인터페이스 개발에 필요한 인력을 구성하며 그룹에 필요한 활동 디렉토리를 지정한다. 그림 10은 협동관리자를 통한 그룹 로그인 창을 보여준다. 그룹 책임자는 기존의 유닉스 그룹으로부터 협동작업을 위해 필요한 구성원을 선택하여 그룹구성원 창으로 이동할 수 있다. 그림 10에서 그룹이름은 DCWA_CASE 명명되어 있으며 활동 디렉토리는 COLABS로 주어졌다.

그림 10. 그룹구성을 위한 협동관리 창

각 그룹구성원은 각자의 사무실에서 DCWA 시스템을 실행시킨 후 원하는 그룹으로 로그인 한다. 로그인이 성공한 후 화상회의 도구, 문서 편집기 및 논리뷰 창을 통해 각 그룹 구성원들은 업무 분담에 관한 토의를 시작한다. 각자에게 분담된 업무가 노드 형태로 창에 디스플레이 된다. 논리뷰의 각 노드들은 할당된 업무에 따라 문서, 그림 및 시뮬레이션 노드로 할당 된다. 또한 구성원들은 CAD 도구를 이용하여 전자우편 시스템 인터페이스의 개략적인 모형을 도출하는 데에 합의 할 수 있다. 그림 11에서 'Drawing' 노드는 그림 노드로 할당되어 있으며, 사용자는 이 노드를 클릭하여 그림 작업을 수행할 수 있다. 한편 그룹 책임자는 논리뷰 상의 문서 노드인 'About' 창을 열어 전자우편 인터페이스 디자인에 필요한 시스템 요구사항 리스트를 작성한다. 이때 역시 각 구성

그림 11. 논리뷰 창(왼쪽)과 그림 및 텍스트 문서 편집기(오른쪽)

원들의 참여가 협동적으로 이루어 질 수 있으며 필요하다면 DCWA 화상회의 도구를 이용하여 그룹 구성원간의 의사 소통을 지원한다. 그리고 구성원들간의 협동작업 시 노드 충돌을 방지하기 위해 특정 노드에 잠금 장치를 이용한다. 이 잠금 장치는 한 사용자가 노드를 사용하고 있을 때 다른 사용자가 그 노드에 접근 할 수 없도록 제한한다[6,17]. 하지만 특정 노드에 잠금 장치가 되어 있어 노드 편집은 불가능하더라도 편집행위의 관찰은 가능하다. 요구사항 리스트에 각 구성원들의 합의사항을 나열한 후 각 구성원들은 할당 받은 업무(e.g., 프로그래밍 업무)를 시작한다. 이때 각 구성원들은 논리뷰의 노드 편집 기능을 사용하여 필요에 따라 노드를 생성, 분기하거나 병합할 수도 있다. 그림 11는 전자우편 시스템 디자인을 위한 전체 문서의 논리뷰가 나무 구조의 형태로 분기되어 있음을 보여준다. 또한 각 구성원들은 작업을 진행하면서 다른 노드의 작업 내용을 검색할 필요가 있다. 사용자는 주어진 검색도구를 사용하여 질의어를 구성함으로써 원하는 노드를 쉽게 검색할 수 있다. 검색된 노드는 세만틱 네트워크로 구성되어 창에 디스플레이 된다. 그림 12는 'help' 노드를 찾기 위한 질의어 창과 결과 노드들을 세만틱 망으로 연결한 세만틱 뷰를 보여주고 있다. 세만틱 뷰는 찾고자 하는 'help' 노드와 의미적으로 연계되어 있는 주위 노드들을 함께 보여줌으로써 노드 검색의 효과를 제고

그림 12. 질의어 창과 세만틱 뷰

할 수 있다.

각 구성원들의 노드 편집이 끝난 후 그룹 책임자는 시뮬레이션 기능을 사용하여 실행파일을 생성하고 시뮬레이션 코드의 이상 유무를 확인할 수 있다. 실행파일에 이상이 있거나 컴파일 에러가 주어지는 경우 해당 노드 책임자는 즉시 코드를 디버깅할 수 있다. 이때 유닉스 MAKE 유틸리티를 이용하여 변경되지 않은 다른 노드의 컴파일링 과정 없이 수정된 노드만을 재 컴파일하여 실행파일을 생성할 수 있다.

표 1. 기존 시스템과 DCWA의 기능 비교

	그룹 컨퍼런싱	CASE	CAD (객체 관계성)	데이터베이스 탐색	동기/비동기
DCWA	비디오/오디오	시뮬레이션	온라인 결합	세만틱 망	동기식+비동기식
MERMAID	비디오/오디오	텍스트편집	-	-	동기식
SPIN	비디오/오디오	텍스트편집	-	-	동기식
CoSARA	텍스트	-	그림 중첩	-	동기식
rIBIS	텍스트	텍스트편집	-	-	동기식
SEPIA	텍스트	텍스트편집	-	-	동기식
TWS	비디오/오디오	-	그림 중첩	-	동기식
GroupPaint	텍스트	-	그림 중첩	-	동기식
Ensemble	텍스트	텍스트편집	그림 중첩	-	동기식
ClearCase	-	시뮬레이션	-	-	비동기식

이상의 사례는 그룹 구성원들의 상호간 업무 조정, 업무 흐름 제어, 독립적으로 작성된 코드의 병합, 테스트 및 시뮬레이션 작업에 DCWA가 효과적으로 이용될 수 있음을 보여준다.

표 1은 기존의 CSCW 시스템에서 구현된 내용과 본 연구에서 제시한 DCWA에서 실제 구현된 내용을 비교하였다.

6. 결론 및 향후 연구과제

DCWA는 텍스트, 그림, 음성, 화상이 결합된 통합 CSCW환경을 제공하며 분산협동 CAD 및 CASE 어플리케이션에 적합하다. 결합된 CAD 및 CASE 도구는 협동작업 관리의 유연성을 증가 시킴으로써 공학 디자인의 작업시간을 단축시킬 수 있다. 또한 시뮬레이션 기능의 지원으로 제품을 생산하기 전에 미리 모형을 시뮬레이션 해봄으로써 제품 개발의 시간과 비용을 단축하는 효과를 가져다 준다. 공학 디자인과 시뮬레이션 기능의 결합은 향후 가상현실, 공학용 하드웨어 인터페이스 및 지식베이스 개발 환경에도 적용될 수 있을 것이다. 실험실의 물리적 제한 때문에 DCWA 시스템은 현재 근거리 통신망 환경에서만 개발되어 있다. 향후에는 인터넷과 같은 원거리 통신망 환경에서도 사용될 수 있는 시스템이 개발 될 것이다. 이때 저속의 통신 속도는 실시간 CSCW 시스템의 구현에 중요한 장애 요소가 될 수 있지만 저속 네트워크 환경에서 특성을 발휘하는 비실시간 CSCW 시스템의 기능들을 접목시킨다면 이 문제에 대한 해결책이 될 수 있을 것이다. 또 다른 이슈로는 다양한 공학 어플리케이션 환경에 적합한 시스템을 개발 하는 것이다. 현재 대부분의 서비스 및 인터페이스는 그룹 구성원간의 경험에 의존한다. 이를 위하여 다양한 어플리케이션에 근거한 서비스 기능 및 인터페이스에 대한 보다 많은 연구가 필요하다.

참 고 문 헌

[1] K. H. Chang, Y. Gong, T. Dollar, S. Gajiwala, B. Lee, and W. Wear, On Computer Supported Collaborative Writing Tools for Distributed Environments, Proceedings of the 1995 ACM Computer Science Conference, pp. 222-229, Mar. 1995.

[2] T. D. Little and A. Ghafoor, Multimedia Synchronization Protocols for Broadband Integrated Services, IEEE Selected Areas in Communications, Vol. 0, No. 9, pp. 1368-1381, Dec. 1991.

[3] T. Znati and B. Field, A Network Level Channel Abstraction for Multimedia Communication in Real-Time Networks, IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No.4, pp. 590-599, Aug. 1993.

[4] H. Ishii and N. Miyake, Toward an Open Shared Workspace: Computer and Video Fusion Approach of TeamWorkStation, Communications

of the ACM, Vol. 34, No. 12, pp. 37-49, Dec. 1991.

[5] T. Liang, H. Lai, N. Chen, H. Wei, and M. Chen, When Client/Server Isn't Enough: Coordinating Multiple Distributed Tasks, IEEE Computer, Vol. 27, No. 5, pp. 73-79, May 1994.

[6] M. Montes, R. Newman-Wolfe, and M. Webb, Implicit Locking in the Ensemble Concurrent Object-Oriented Graphics Editor, Proc. of the 1992 CSCW Conference, pp. 265-272, Nov. 1992.

[7] G. L. Rein and C. A. Ellis, rIBIS: A Real-Time Group Hypertext System, Int. Journal of Man-Machine Studies, pp. 339-367, Aug. 1993.

[8] IEEE, IEEE Communications Magazine, Vol.32, No.5, May 1994.

[9] J. Rosenberg, R.E. Kraut, E.L. Gomez, and C.A. Buzzard, Rendezvous: An Architecture for Synchronous MultiUser Applications, Tech. Report TMARH016223, Bellcore, May 1992.

[10] R. Steinmets, Synchronization Properties in Multimedia Systems, IEEE Selected Areas in Communications, Vol.8, No.3, pp. 401-412, Apr. 1990.

[11] A. W. Wear, Y. Gong, and K. Chang, Database Management for Multimedia Distributed Collaborative Writing, Proc. of the 33rd ACM Southeast Conference, pp. 42-51, Mar. 1995.

[12] R. C. Schank, Conceptual Information Processing, Amsterdam, North Holland, 1975.

[13] R. S. Fish, R. E. Kraut, M.D. Leland, and M. Cohen, Quilt: A Collaborative Tool for Cooperative Writing, Proc. of the Conference on Office Information Systems, pp. 30-37, Palo Alto, CA, 1988.

[14] S. Opper, A Groupware Tool Box, BYTE, pp. 275-282, Dec. 1988.

[15] Liberty, Jesse and J. Mark Hord, Teach Yourself ANSI C++ in 21 Days, MS Publishing, pp. 609-621, Indianapolis, IN, 1996.

[16] Mark Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning, L. Suchman, "Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings, CACM, Vol. 30, No. 1, pp. 33-47, 1995.

[17] Byong G. Lee, Design and Development of Distributed Collaborative Writing Aid and its Integration, Auburn University, Dept. Computer Science and Eng., M.S. Thesis, Aug. 1996.

[18] Timothy W. Dollar, A method for maintaining spatial relationships among graphical objects in a distributed collaborative writing environment, Auburn University, Dept. Computer Science and Eng., M.S. Thesis, Mar. 1995.

[19] M. Watae, S. Sakata, K. Maeno and T. Ohmori, Distributed Multoparty desktop conferencing system: MERMAID, In Proceedings of the Conference on Computer Supported Cooperative Work, CSCW 90, pp 27-38, ACM, New York, 1990

[20] S. Berson, G., Estrin, Y. Eterovic, I. Tou, and E. Wu, Prototyping Synchronous Group Applications, Conputer, IEEE, Vol. 27, No. 5, pp. 48-56, May, 1994.

[21] Http://www.pureatria.com/products/clearcase/

[22] K. Ehrlich, Designing groupware applications: A work-centered aproach., In Beaudouin-Lafon (Ed.), Comnputer-Supported Cooperative Work, Trends in Software Series 7:-28, John Wiley & Sons, Chichester, 1999.

[23] C. Ellis, Workflow technology, In Beaudouin-Lafon (Ed.), Comnputer-Supported Cooperative Work, Trends in Software Series 7:29-54, John Wiley & Sons, Chichester, 1999.

이 병 걸

1988년 University of Bridgeport 물리학과 (학사)
1996년 Auburn University 전산학과 (공학 석사)
1998년 Auburn University 전산학과(공학 박사)
1990년~1991년 시스템공학센터 연구원
1998년~현재 서울여자대학교 컴퓨터학과 조교수
관심분야 : 소프트웨어 형상관리, CSCW, 분산시스템